MORRISON & FOERSTER LLP
MICHAEL A. JACOBS (Bar No. 111664)
mjacobs@mofo.com
MARC DAVID PETERS (Bar No. 211725)
mdpeters@mofo.com
DANIEL P. MUINO (Bar No. 209624)
dmuino@mofo.com
755 Page Mill Road, Palo Alto, CA 94304-1018
Telephone: (650) 813-5600 / Facsimile: (650) 494-0792

BOIES, SCHILLER & FLEXNER LLP
DAVID BOIES (Admitted *Pro Hac Vice*)
dboies@bsfllp.com
333 Main Street, Armonk, NY 10504
Telephone: (914) 749-8200 / Facsimile: (914) 749-8300
STEVEN C. HOLTZMAN (Bar No. 144177)
sholtzman@bsfllp.com
1999 Harrison St., Suite 900, Oakland, CA 94612
Telephone: (510) 874-1000 / Facsimile: (510) 874-1460

ORACLE CORPORATION
DORIAN DALEY (Bar No. 129049)
dorian.daley@oracle.com
DEBORAH K. MILLER (Bar No. 95527)
deborah.miller@oracle.com
MATTHEW M. SARBORARIA (Bar No. 211600)
matthew.sarboraria@oracle.com
500 Oracle Parkway, Redwood City, CA 94065
Telephone: (650) 506-5200 / Facsimile: (650) 506-7114

*Attorneys for Plaintiff*
ORACLE AMERICA, INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

| | |
|---|---|
| ORACLE AMERICA, INC.<br>Plaintiff,<br>v.<br>GOOGLE INC.<br>Defendant. | Case No. CV 10-03561 WHA<br>**ORACLE AMERICA, INC.'S REPLY IN SUPPORT OF MOTION TO AMEND '205 PATENT INFRINGEMENT CONTENTIONS AND SUPPLEMENT EXPERT REPORTS**<br>Dept: Courtroom 8, 19th Floor<br>Judge: Honorable William H. Alsup |

After Oracle filed its motion to amend its infringement contentions and supplement its expert reports with respect to the ’205 patent, the Court set the case for trial to begin on April 16. (ECF No. 786.) It does not appear likely that the ’205 examiner will address the ’205 patent further before the patent phase of trial begins. Today, Oracle files its response to the ’205 examiner’s rejection, and the deadline for Google to file its comments is April 18. Given Oracle’s offer to dismiss with prejudice the claims of the ’205 patent that remain rejected as of the time of a spring trial (ECF No. 777 at 3), the Court may well decide to deny this motion as moot. If the Court does so, it should be without prejudice. If, for some unforeseen reason, the trial does not go forward in the spring, the yardsticks by which diligence and prejudice are measured change substantially, and Oracle should be permitted to re-file its motion.

If the Court wishes to address the merits of Oracle’s motion at this time, it should allow Oracle to amend its infringement contentions and supplement its expert reports to identify the specific virtual machine instructions that are executing when Android dexopt tool generates new virtual machine instructions, thereby infringing the asserted claims of the ’205 patent. Oracle’s amendments do not alter its theory that dexopt’s inline substitution functionality infringes the asserted claims of the ’205 patent. Oracle’s theory has always included the time when dexopt is running. The infringement contentions disclose that dexopt can be run “at runtime” by the Android installer as well as in a “just-in-time” fashion on a development device.

Google’s selectively quotes from Prof. Mitchell’s reports to remove context and argue that what he said means something different. For example, Google quotes (using bold italics, no less) Prof. Mitchell’s statement that Android’s dexopt obtains needed information “***by initializing a Dalvik VM and loading all the classes from an application’s .dex file into the VM.***” (Google Opp. at 2-3.) That is a true statement. But it has nothing to do with whether the ***application*** is running. Neither does the full paragraph from which Google plucked its sentence:

> Prof. August fails to consider the evidence that dexopt runs at runtime. As Google wrote: “[t]he solution is to invoke a program called dexopt, which is really just a back door into the VM. It performs an abbreviated VM initialization, loads zero or more DEX files from the bootstrap class path, and then sets about verifying and optimizing whatever it can from the target DEX. On completion, the process exits, freeing all resources.” (*See* http://android.git.kernel.org/?p=platform/dalvik.git;a=blob;f=docs/dexopt.html.)

> The optimizations that dexopt performs (such as replacing symbolic references with vtable offsets) are ones that require information that is only available at runtime (such as a vtable offset), and that information is obtained by initializing a Dalvik VM and loading all the classes from an application's .dex file into the VM. Google confirms it: "[s]ome of [the optimizations performed by dexopt] require information only available at runtime, others can be inferred statically when certain assumptions are made." (*See id.*) The way Google wrote dexopt, rewriting the DEX method invocation instructions with EXECUTE_INLINE and EXECUTE_INLINE_RANGE instructions is one of the former kind of optimizations.

(Francis Decl. Ex. B (Mitchell Reply Report) ¶ 60.) Prof. Mitchell's argument here is that one can tell that dexopt runs at runtime because it requires information available only at runtime, as Google admits in its dexopt documentation. He did not argue that dexopt runs at runtime because it then proceeds to run the application it just optimized. Prof. Mitchell acknowledges the opposite, in fact, when he quotes Google's dexopt documentation to the effect that dexopt loads the application into a Dalvik VM but then exits upon completion of the optimization. (*Id.*)

Google's quotations from Prof. Mitchell's deposition likewise take his testimony out of context and—worse—fail to quote testimony that directly contradicts its argument. With respect to the testimony that Google did quote, Google's elisions disguise the fact that Prof. Mitchell was not testifying with respect to the '205 patent about dexopt running on an Android phone but was instead testifying with respect to the '104 patent about an experiment he performed using Android's development environment in which he compiled an application and then installed and ran it on the Android SDK's emulator:

> Q. BY MR. KAMBER: Professor Mitchell, I just want to ask a few questions to finish up on the '104, because we need to move to the '205 patent. You would agree with me that the output of running dexopt on a .dex file is a .odex file; correct?
>
> A. I believe that may be one option. When I installed the SDK and used this to run code, I didn't generate an output like that. It's just part of the process of running the application that I compiled and translated to Dalvik. And then the run process of executing that in the Dalvik Virtual Machine, I believe, was what -- where the dexopt got invoked.
>
> Q. So did you only run this by way of the SDK?
>
> A. That's the -- an experiment that I did. Anytime you run an Android phone, you may also run this, so that's another example.

(Francis Decl. Ex. C (Mitchell Dep.) at 342:7-22 (portions quoted by Google highlighted).) (The word "this" in the second answer refers to dexopt, which indeed may run anytime one runs an Android phone, according to the source code in PackageManagerService.java.)

The testimony that Google should have quoted in its opposition is when Prof. Mitchell testified that dexopt's "install time" qualifies as "runtime," and that an example of why that is so could be found in the portion of his report that discusses the '205 patent:

> Q. How did you determine that dexopt was at runtime?
>
> A. Through statements from Google describing the way that this operates. There are slides, again, and quotations and sections from documentation, I believe. And also looked at the code and understood how this behaves and the nature of the steps and the information required in order to complete those steps.
>
> Q. Let's look at some of that documentation. Take a look at page 90. You have a clip of a slide and then a quote from Mr. Bornstein's presentation here on page 90; correct?
>
> A. This certainly is a slide and a quote. I believe it's from one of Dan Bornstein's presentations. It may be the one we identified earlier.
>
> Q. And the quote, you say that -- you give at 21:54 under, quote, "install time work." Do you see that?
>
> A. Yes.
>
> Q. Okay. So what Mr. Bornstein is talking about that you cite to relates to install time on a device; correct?
>
> A. That's the way he's characterized this portion of the operation, yes.
>
> * * *
>
> Q. Why do you think that this work that's being done at install time qualifies as runtime under the terms of the patent?
>
> A. I think what Dan Bornstein is trying to emphasize is things that are done once. Although, there are other documentation information that's explicit on the runtime nature. I believe, for example, the documentation that I reproduced on page 178 and 179, that explains it in slightly more technical terms.
>
> Q. Well, what you said on 178 and 179 relates to the '205 patent; correct?
>
> A. It's presented there. It's an explanation of how -- it's an explanation of dexopt from the Android website. It's the same dexopt in both patents. I don't know if it's also -- that same section is also cited here in connection with '104. That's just one place that I knew I had put that information. So that's why I pointed to it now.

(Peters Supp. Decl. Ex. D (Mitchell Dep.) at 329:25-332:12.) Note that paragraph 402 of the Mitchell report (which spans pages 178 and 179) includes the paragraph about the installer that

Google quotes on page 3 of its opposition. (*Compare* Google Opp. at 3 *with* Francis Decl. Ex. A (Mitchell Report) at ¶ 402.) There can be no question that Prof. Mitchell advanced an infringement theory based on the "install time work" performed by dexopt, because it is right there in his testimony and his report. Oracle's original contentions, which identify dexopt as performing the step of "generating, at runtime, a new virtual machine instruction" when it performs inline substitutions, likewise quote Google's admission that dexopt performs these optimizations "either in the build system or by the installer. On a development device, dexopt may be run the first time a DEX file is used and whenever it or one of its dependencies is updated ('just-in-time' optimization and verification)." (Peters Decl. Ex. A at ¶ 13.) In its infringement contentions, Oracle accused non-build-system, non-development devices (such as the Droid) of infringement—for these devices, how could Google reasonably believe that Oracle was basing its dexopt infringement theory on anything ***but*** the installer?

In its opposition, Google fails to address the key fact that pops the balloon of its argument: the Android installer, which runs dexopt, is itself an application that runs in a Dalvik virtual machine. As Prof. Mitchell lays out in detail in his supplemental report, the virtual machine instructions of the installer are the ones executing when dexopt optimizes an application during its installation. (Peters Decl. Ex. B at 8-14.) As Google knows better than anyone, and as Prof. Mitchell testified, "install time" on an Android device is runtime. This is why Google's documentation (disclosed in both Oracle's infringement contentions and Prof. Mitchell's reports) states that dexopt requires information "only available at runtime." (Francis Decl. Ex. B (Mitchell Reply Report) ¶ 60.) In truth, Google's argument about what Oracle's position "has always been" is based on either Google's misunderstanding or wishful thinking colored by its non-infringement position, rather than on an objective assessment of Oracle's infringement contentions.

As stated in its opening brief, Oracle has good cause to amend because the Court's construction of "at runtime" was not only different from Oracle's proposal but also quite different from Google's ambiguous construction. Oracle provided the proposed amendment and supplemental reports to Google about a month after the Court construed the phrase "at runtime."

Oracle did so in less time than other parties did that were found to be diligent. *Schindler Elevator Corp. v. Otis Elevator Co.*, No. 06 Civ. 5377 (CM)(THK), 2010 U.S. Dist. LEXIS 110313, at *11-12 (S.D.N.Y. Oct. 6, 2010) (allowing plaintiff's expert to disclose revised infringement theory seventy-four days after issuance of claim construction order because expert "needed some time to digest the court's rulings and revisit the file wrapper"). If, for some reason, the trial does not take place in the spring, Google will not be prejudiced by the Court allowing this amendment, for Google will have ample time to prepare its non-infringement analysis and expert report.

With respect to Google's request for leave to amend its invalidity contentions, Google has not provided the Court or Oracle with its proposed amendments or even specifically identified the prior art that would be included in the amendment.[1] Oracle cannot evaluate whether the Court's construction of "at runtime" would justify Google's amendment under Patent Local Rule 3-6 or whether Oracle would be prejudiced by it.

Dated: March 19, 2012

MICHAEL A. JACOBS
MARC DAVID PETERS
DANIEL P. MUINO
MORRISON & FOERSTER LLP

By: /s/ Michael A. Jacobs

*Attorneys for Plaintiff*
ORACLE AMERICA, INC.

[1] Google's mention of a Deutsch & Schiffman publication is particularly hard to understand. Google submitted a '205 anticipation contention based on a Deutsch & Schiffman article in its original invalidity contentions in January 2011, but withdrew the contention in June 2011. Google moved the Court to amend its invalidity contentions to add an obviousness contention based on a combination of the Deutsch & Schiffman article with a Magnusson article in July 2011, which the Court granted (ECF No. 281 at 7-8), but Google withdrew the contention in October 2011. (ECF No. 475.) Google submitted a claim chart based on the Deutsch & Schiffman article in the *inter partes* reexamination, but the examiner concluded that the article did not raise a substantial new question of patentability, and so it was excluded from the reexamination. Google may have a different Deutsch & Schiffman reference in mind.